Faculdad de Informática
Universidad Politécnica de Madrid

Dental Implants Success Prediction: Tomography Based and Data Mining Based Approach

Master of Science Thesis

Author : Abdallah ELALAILY
Supervisors : Prof. Consuelo Gonzalo
Prof. Ernestina Menasalvas
Submission date : 11 September, 2013

Abstract

There are a number of factors that contribute to the success of dental implant operations. Among others, is the choice of location in which the prosthetic tooth is to be implanted. This project offers a new approach to analyse jaw tissue for the purpose of selecting suitable locations for teeth implant operations. The application developed takes as input jaw computed tomography stack of slices and trims data outside the jaw area, which is the point of interest. It then reconstructs a three dimensional model of the jaw highlighting points of interest on the reconstructed model.
On another hand, data mining techniques have been utilised in order to construct a prediction model based on an information dataset of previous dental implant operations with observed stability values. The goal is to find patterns within the dataset that would help predicting the success likelihood of an implant.

Contents

Chapter 1

Introduction

The idea of this study is to find a method that helps predicting the degree of success of an implant in a certain patient, in a way that would increase the likelihood success of such operations.

After examining the problem at hand, it was decided that this research integrates two approaches. The first one, examining jaw medical imagining scans to determine what information can be extracted from there. And the second involving data mining, in order to look for patterns within records, of already performed operations, to determine if there are in fact patterns within patient variables that increase or decrease the likelihood of success for dental implant operations.

Over the years, technology improvement within the medical imaging domain has created more thorough and complex data, that it created the need for computational systemic ways of extracting information from such images. Biology research thus relies heavily on computer science advances in order to approach research problems.

The main idea of the research is to automate some analysis techniques to extract information from medical imaging slices in a way that could help information preparation for dental implant operations or other relevant dental applications. To do so, we had to get acquainted with the medical aspect of the operation, and the rationale behind further studies on the topic. This was conducted in order not to steer away from the practical applications of the case we were asked to assist with.

Data mining was then thought of as a technique that would create a predictive model for patients undergoing dental implant operations. Data mining would be used to extract on one hand clusters of patients in which implants behave similarly, and on the other finding those discriminative factors that may cause the implant operation either success or failure.

1.1 Dental implants

In dentistry, dental implants are root devices used to replace one or more missing teeth in the human jaw. Dental implants are bolts placed in the human jaw to replace missing teeth. They are normally of metallic structure, usually titanium, made in a way to react with human jaw bone structure so that after healing it becomes difficult to separate from the natural tissue. The process by which this occurs is called osseointegration. If a dental implant becomes osseointegrated, the implant operation is considered a successful one. Osseointegration is affected by a number of patient and operation variables. This is further discussed in the following chapter.

1.2 Computed tomography

X-ray computed tomography is the use of X-rays to acquire tomographic images of specific body parts. The process is used for diagnostic purposes across numerous medical disciplines. Computed tomography produces data that could be manipulated in numerous ways to show different physical structures based on individual body parts absorption of the X-rays. The tomographic images are then used to reconstruct a three dimensional representation of the scanned volume. Aside from medical applications, computed tomography also has other applications such as material testing, and archaeological scanning.

1.3 Objective

The main goal of this study is to implement computational techniques that will further assist dental implant operation planning. Doing so by integrating two approaches. One which dental medical scan files are analysed for information that could be deemed beneficial. While the second is to analyse

patient related information as well as dental implant parameters to determine factors that increase or decrease the likelihood of success of dental implant operations.

1.4 Thesis structure

This thesis consists of 6 chapters, including this one. The following chapters are, *Background*, *Approach*, *Implementation*, *Results & Discussion* and *Conclusion & Future Work* respectively. The *Background* chapter reviews contextual information on the technology and software used for the research project this document covers. Then, the *Approach* chapter explains the development followed for addressing the research problem. Next, the *Implementation* chapter examines the algorithms developed. After that, the *Results & Discussion* chapter points out analysis of the results achieved following the approach and implementation methodology discussed in the previous chapters. And finally the *Conclusion & Future Work* chapter covers the implications of the research findings and sheds light on the direction of some possible future work that builds upon the findings of the research.

Chapter 2

Background

This chapter discuses the background of concepts and technologies that are used or mentioned throughout this research study.

2.1 Dental implants

2.1.1 History

On a historic level, evidence suggest that dental implants may have dated back to the Ancient Egyptian as early as 3000 B.C.[1][2]. Evidence for dental implants were found in Mayan civilisation artifacts that dated back to 600 A.D.[2]. Middle-Age Europe, for up until the 17th century, has seen practices were human teeth were bought off from the poor or corpses to be allotransplanted to other individuals[2]. This practice however involved the risk of spreading infectious diseases.

However, on a medical level, dental implants started taking place around the 1800s, the beginning of the modern surgery era, with concepts of sterilising and disinfecting surgical apparatus[2]. But in any case, dental implant operations in a clinical sense was introduced in 1918[3]. The techniques of the dental implant operations have changed since then. Modern day dental implants rely on a technique discovered by Per-Ingvar Brånemark, while conducting experimentation in which titanium implant chambers were used to study flow of blood in rabbits' bones. After however it was time to remove the chambers from the bone, he realised that bone tissue had completely fussed with the titanium chamber that it has become impossible to remove[4]. Brånemark then considered the possibility for human use, and

went on to study the effects of titanium implants and bone integration. He proved that bone tissue could integrate with the titanium oxide layer of an implant that it becomes impossible to separate without fracturing the surrounding bone[5]. Brånemark named that process *osseointegration*, derived from the Greek word *osteon* (bone) and the Latin word *integrare* (to make whole).

An implant is said to have osseointegrated when the implant is not able to show relative movement between its surface and the bone it is in contact with. In essence the objective is to achieve a stabilising mechanism where the prosthetic implant can be incorporated into bone tissue and last under normal conditions of loading[6].

2.1.2 Factors affecting osseointegration

Several factor may affect osseointegration favourably or unfavourably. Factors that have a favourable effect include implant related factors, such as; implant design, chemical composition, topography of the implant surface, material, shape, diameter, etc. Other factors relate to the implant site, the host bone tissue and its ability to heal. Furthermore, loading conditions and the use of adjuvant agents[6].

While factors that might have an unfavourable effect include, excessive implant mobility and factors that contribute to that, mismatched coating for the implant, undergoing radiation therapy, and patient related conditions such as; osteoporosis, rheumatoid arthritis, nutritional deficiency, smoking, etc.[6].

2.2 Computed tomography scanning

The use of computed tomography scanning has been cemented in radiology diagnostics soon following its introduction. The technology was conceived in the mid 1960s. Sir Godfrey Hounsfield thought of an experimental project in which he attempted to reconstruct contents of a box from different readings at randomly assigned directions. He used a programme to attain absorption values collected, then used another programme to reconstruct those values into a three-dimensional model. Much to his surprise, Sir Hounsfield, found the results to be much more accurate than he expected[7].

Hounsfield describes the technology as a scanning mechanism in which X-ray readings are recorded from a number of angles. The data is then translated to absorption values and displayed as a number of pictures (slices). The technology is far more accurate than traditional X-ray systems in a way that differences in tissues of almost the same density can be highlighted. He then proceeds to describe how the same X-ray technology used in the conventional way, then, losses precision. Through transforming all information acquired from a three-dimensional body onto a two-dimensional film, where the image overlaps objects from back to front. And for which an object to be noticeable, it has to be very different from all the objects that are positioned forward and backward relative to it[8].

Shortly after the introduction of computed tomography scanning technology, scan data were taken on tape for processing on an ICL 1905 computer, that required around 20 minutes per slice. Often the reconstruction would be left as an overnight process. The process was thought of as so complicated that it would require mainframe processing abilities. However this all changed as computer technology improved and reconstruction algorithms became more efficient[7].

2.3 Image processing

Digital image processing is the process of using computer algorithms to enhance/show details on raw digital images. The output of image processing is either an image or image characteristics set that is associated with the input image. Image processing maybe be performed for number of reasons:

Image restoration Improving the quality of an image.

Object recognition Detecting different objects in an image.

Pattern recognition Detecting patterns in an image.

Image retrieval Seeking image(s) of interest.

Visualisation Displaying objects that are not clearly visible in an image.

The applications of image processing are numerous, and can range from textile manufacturing monitoring to military surveillance operations. But what the diverse applications have in common is the need for storing, preprocessing and then interpreting the raw images.

2.3.1 Fiji

Fiji[9] is an open-source image-processing software project that is based on the skeleton of ImageJ[10]. Fiji addresses problems that underlie with ImageJ. For ImageJ is a tool that was developed for biological research by biologists. The result is a software architecture that does not adhere to modern software conventions. Fiji, however, readdresses the outdated architecture, while simultaneously introducing software libraries that deploy a number image analysis algorithms.

Fiji is still compatible with ImageJ, while adding extended functionality. In essence, Fiji serves as a distribution of ImageJ that includes additional libraries and plugins specifically targeting biological image analysis[9].

2.4 Data mining

Knowledge Discovery from Databases (KDD) is defined as the non-trivial technique for processing and understanding raw data, and in doing so transforming it into data that can be used to give actionable knowledge. The term KDD was put forward in 1989, and it has become prominent in the fields of artificial intelligence and machine learning since then. To find patterns in data, it incorporates various fields of knowledge such as, statistics, database knowledge, machine learning and visualisation. The process includes data cleaning, data storing, machine learning algorithms, processing and visualisation.

There are two types of problems, descriptive problems and predictive problems. Descriptive problems are problems which data within a dataset is to be interrupted and illustrated in the form of patterns, without the need of any forecasting. While predictive problems are problems which models are designed for information in a dataset in order to anticipate attributes. The aim of which is to predict an attribute (target variable) based on analysed patterns in other attributes. For the purpose of this study a predictive model is needed, and as such classification techniques are utilised to achieve that target.

2.4.1 CRISP-DM

Data mining process is a far more complex process than applying machine learning algorithms on a dataset. It is a gradual process that requires information acquisition and knowledge understanding beforehand. As that is the case, for the data analytics in this study, the CRISP-DM methodology (*Cross Industry Standard Process for Data Mining*) was followed. CRISP-DM is the most used process for data mining projects, and often referred to as the "de facto standard". CRISP-DM draws guidelines to the phases of development that is to be followed for a data mining project. It also sheds light on the details and deliverables of each phase[11].

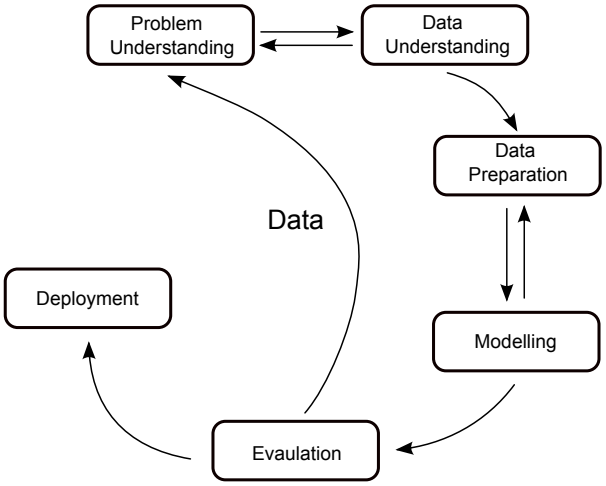

Figure 2.1: The phases of CRISP-DM standard model process.

There are 6 phases in the CRISP-DM methodology:

Problem Understanding

The initial phase which focusses on understanding the project, its objectives and from the business point of view. This includes thorough comprehension of different business related aspects, and how they would relate to one another. The knowledge attained in this phase results in drafting the data mining problem and having a preliminary plan in order to achieve targeted goals.

Data Understanding

This phase aims at comprehending dataset attributes and how they

relate to the problem. It starts with initial data collection. This is followed by further inspection of the dataset to better one's understanding, discovering connections and forming an early hypothesis.

Data Preparation
A phase that involves cleaning the initial raw data in order to get to a formal dataset to work on for the later phases. Data preparation is likely to be performed multiple times, as in some cases the data formatting or selection is constricted by the choice of modelling algorithms.

Modelling
In this phase various machine learning and data mining algorithms are applied to the dataset. Some algorithms may have specific requirements regarding the data form. As such, the data preparation phase may be revised.

Evaluation
Before proceeding to the final model deployment, the model is assessed thoroughly and is checked to be in compliance with the predetermined objectives. And in doing so reviewing the steps executed to construct the model.

Deployment
Finally, after passing the evaluation phase, the model is deployed and results are collected. The creation of the model is generally not the end of the project. Even if the purpose was to increase knowledge of the data, information will be organised and presented in a way that the end user can use.

2.4.2 Classification analysis

Classification analysis deals with defining a set of categories on data within a dataset whose attributes are known, in other words. The attributes are analysed and quantised to allow basis for the classification. The purpose of learning algorithms is to create a classifier given a labelled training set.

Classification analysis not a very exploratory technique. It serves more of assigning instances to predefined classes, as the objective is to figure out a way of how new instances are to be classified. It is considered as a form of supervised learning, in which a dataset of correctly identified instances is given. Some clustering algorithms only work on discrete attributes, and as such would require preprocessing on numeric based attributes to make them

discrete.

Classification analysis is one of pattern recognition algorithms, that serve to assign a kind of output label to given input values. It is a typical approach in data analysis applicable in numerous fields, such as, machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

2.4.3 Weka

Weka is a machine learning software environment developed at the University of Waikato, New Zealand in Java. Weka was developed as it was observed that most research in the field of machine learning was focused on the renovation of new machine learning algorithms, whilst little attention was given to expanding existing algorithms to real problem applications. The development was focusing on allowing the use of multiple machine learning techniques that has a simple to use interface with visualisation tools, while also allowing some pre- and post-processing tools, with the goal of reaching support for scientists in various fields, and not just machine learning experts[12][13].

Chapter 3

Approach

This chapter discusses the process that led to the development and implementation of the application as it currently stands.

3.1 Problem

Formally characterised, this study focuses on integrating two approaches with the main goal of finding a methodology that helps predicting the success of dental implant operations. The first of which the image analysis of the medical image files of the jaw area. The idea being, taking as input a DICOM (*Digital Imaging and Communications in Medicine*) file stack, and performing investigation based on tissue density differences computed from those files. So as to find information that can be integrated together with the structured information obtained from patients. This information will be later used by a knowledge extraction process. Consequently the second approach is to apply data mining techniques to find patterns from the integrated information. This is done with the objective being creating a model that would be able to calculate the success likelihood of a future dental implant operation based on the variable parameters and how they would fit in the patterns detected from the dataset analysis.

3.2 Jaw computed tomography slice analysis

3.2.1 Visualisation

The first aspect of the problem is the input, how to handle the information files. Input files in this case are multiple DICOM files per application run.

Computed tomography produced volumes of data are windowed around the jaw area. Each file contains a two dimensional horizontal scan of the skull, known as a *slice*. Slices are then stacked on top of one another bring up three dimensional volumetric data. From there it is possible to render a three dimensional volume of the scanned body part.

Fiji plug-ins are deployed for the purpose of three dimensional rendering and visualisation of medical scans in this project.

3.2.2 Excess volume cropping

The next aspect is extracting a region of interest from the windowed scan files as the entire scan volume contains information that is not needed. The regions of interest in this case are the locations of individual teeth in the jaw. Any additional tissue that is not trimmed from the scan volume is an unnecessary inefficiency. As it increases the computations throughout the run of the application. And thus it is in the best interest to minimise, as much as possible, the unimportant tissue from the application's perspective.

To deal with this, we thought of doing analysis based on the digital value of DICOM files. The digital values correspond to the density of the scanned tissues. The idea being, teeth are the most dense tissue in the entire human body. As this is the case, scanning an image or a stack volume for the average most dense tissue would segment the jaw area (containing teeth) from the rest of the scan.

The implementation idea was first tested for two dimensional images, to see if it holds. Given a two dimensional slice, a box of fixed size traverses throughout the image, in doing so, calculating the average density at this specific location. The location where highest average density is located is stored till all average densities are calculated. After that the region with the highest average density is highlighted.

As *figure 3.1* shows, small-sized average boxes do manage to successfully highlight the jaw area, which is our region of interest. However as boxes' sizes increase, the highlighted region switches from the jaw area, to a bit back in the slice. This can be observed clearly in the swtich from an average box size of 250×100 *(c)* to 300×125 *(d)*. This takes place because a bigger box engulfs a lager null area, as the jaw has a relative smaller width than the rest of the head on the slice. Despite the relative high density of teeth, this phenomena disallows a big average box to highlight our region of interest,

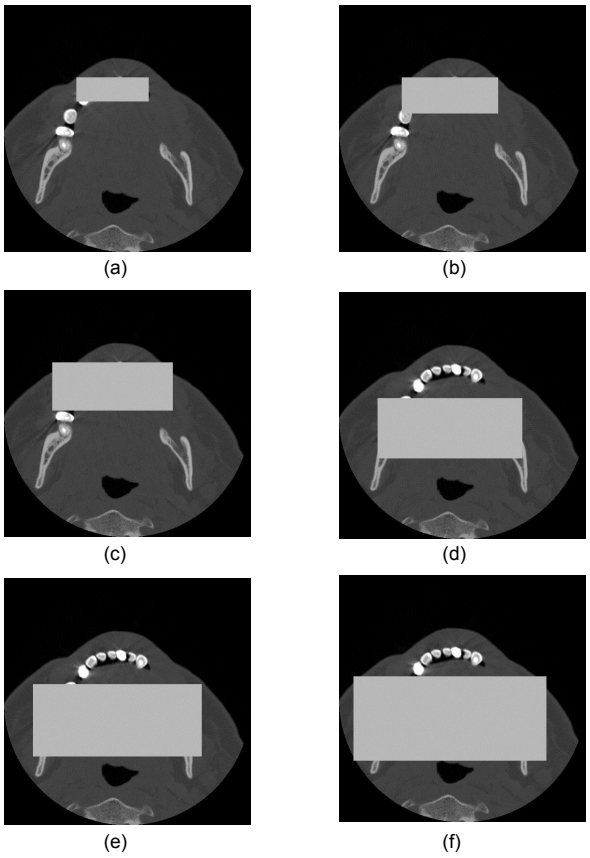

Figure 3.1: Given a 512×512 slice, the highest average density is highlighted using average boxes of different sizes; (a) 150×50, (b) 200×75, (c) 250×100, (d) 300×125, (e) 350×150 and (f) 400×175

and instead, highlighting an area that has a more or less monotonic lower density but that does not have null values indicated by the absence of tissue.

This seemed problematic at first, as the technique did not yield the expected result. However, another approach came to mind.

Instead of aiming directly for highlighting the region of interest, the direction was changed to use the average calculating box to determine the point after which lies the unwanted region. The aspect ratio of the box was changed, increasing the width to height ratio. And thus, the box would locate the widest region of the scan that has a more or less monotonous density values, and would not select the jaw region as a thin wide box would engulf a

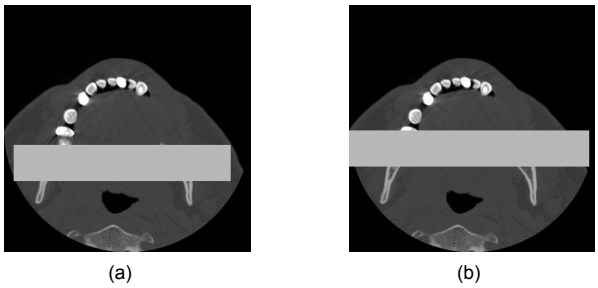

Figure 3.2: Alternative approach using a box with bigger width-to-height ratio in order to mark the monotonous area, which is located directly below the region of interest. Again given a 512×512 slice and box sizes (a) 450×75 and (b) 500×75.

lot of null values at the narrowest region of the scan which corresponds to the jaw. *Figure 3.2* shows the results achieved by using the previously described box in order to differentiate between rejoins in a jaw slice scan based on the average calculated density.

From there, eliminating the unwanted area becomes a trivial problem of figuring out the orientation of the slice, leaving the jaw area in place while removing the rest of the tissue below, as shown in *figure 3.3*.

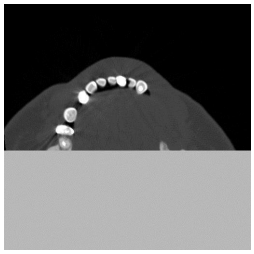

Figure 3.3: Previous technique deployed to eliminate unwanted region of a slice.

The following step would be to extend this technique to the three dimensional stack, as opposed to the two dimensional primary implementation previously discussed.

There were a number of considerations to extend this technique into one more dimension. The first of which how many degrees of freedom should the average calculating cube be allowed to have. The two dimensional box had fixed dimensions based on the best results achieved through the testing process. How would adding a new dimension to this affect the outcome was still unknown.

It was decided the cube would span the entire stack, and as such be only allowed one degree of freedom. The cube has the width of an entire slice, the height of the entire stack and only moves along the y-axis evaluating the average density. The results achieved were fairly predictable. The three dimensional extension of the two dimensional implementation achieved similar results to those achieved running the two dimensional implementation. The same points for extraction were reached in both cases.

But while this removes nearly half of the stack volume, depending on scanning variables as well as the windowing used, this wasn't enough. There still remains unnecessary tissue in the stack. There needs to be further cropping to remove the part of the stack which includes the nose in some cases, or the area above the jaw generally speaking. However this was not a simple task, as some of the scans are upper jaw scans while others are lower jaw scans. Deciding which part to remove next would have to rely on figuring out which type of scan the application is dealing with and acting accordingly.

Upper jaw scans are windowed along with the nose tip, while lower jaw scans are windowed along with the chin. Examining test data indicated that the region of interest is always contained in one half of the remaining scan.

If the scan is of the upper jaw, then the region of interest is in the lower half of the scan, as the upper half contains the nose and middle skull tissue. And if however the scan is of the lower jaw, then thee region of interest is in the upper half of the remaining volume, as the lower jaw is windowed with the chin and lower skull tissue. The problem then became coming up with a technique that would figure out which remaining half of the volume to remove.

A simplistic approach was developed for that sub-problem. Finding the number of high value voxels in the remaining scan, the half-volume that contains higher number of dense voxels is the half-volume that contains the region of interest. This stems from the fact that teeth are the highest density tissue in the human body. And as such, given that the region of interest (jaw)

contains at least one tooth, the points with higher density are located there, and the other half of the scan is eliminated. *Figure 3.4* shows the results of this approach.

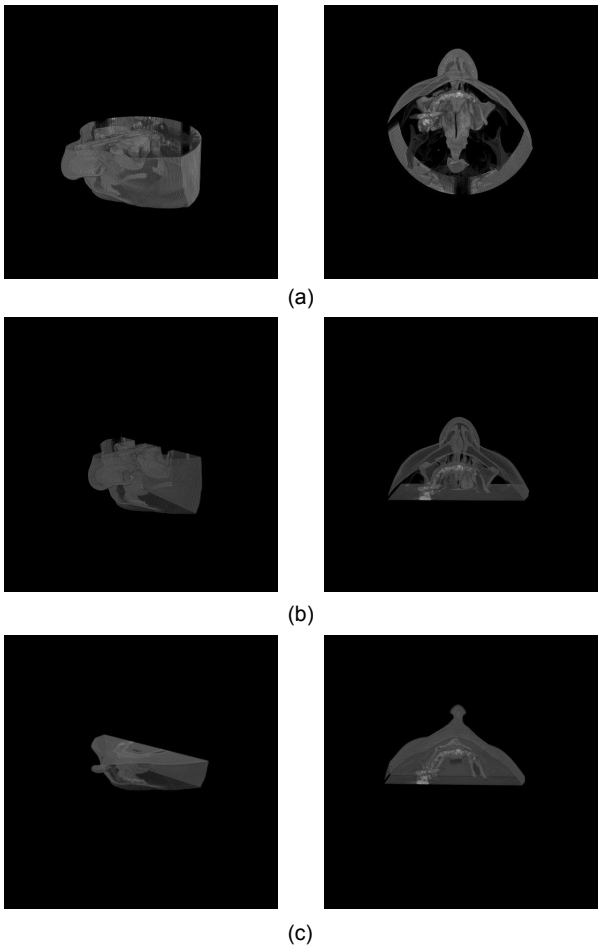

Figure 3.4: Two views of the three dimensional reconstruction of the entire stack, (a) before cropping, (b) after removing the back end of the skull (average density cube), and (c) after eliminating further half of the remaining volume (point of highest intensity).

3.2.3 Point selection

After the original scan has been trimmed of unneeded data, there comes the point where individual teeth needs to be marked to allow further computations. And while it seems like a trivial task of allocating clusters of high density voxels, the task is presented with challenges.

There are a number of factors to consider while planning an algorithm that performs such a task. Often scan imperfections distort the given data. As a result, two or more teeth could be connected and this from a computational view point be observed as one body after the merging. As well as some voxels of high intensity have been observed in areas where they do not belong. Furthermore, the lack of uniformity of the space occupied by teeth in three dimensional space, caused by different teeth types, positions and orientations, further complicates the task of tooth marking.

The primary purpose of the tooth selection technique was not solely to allocate positions of teeth. But the intention was also to allocate locations where teeth used to reside. The rationale was that the site in with a tooth's root is fixed to, is of very dense material too. And that those locations will be highlighted if an algorithm is searching for highly dense voxels. However repeated tests on different scan cases have concluded that this is not the case. Only the enamel of teeth gets highlighted when looking for highly dense tissue in the medical scan. While searching for fairly dense tissue results results in highlighting a lot of redundant tissue, that it makes it impossible to pinpoint a base of a tooth.
After it was confirmed that the selection of both teeth and bases of teeth was not computationally feasible at the moment, it was decided to settle for highlighting only location of teeth based on the enamel density highlighting.

Much consideration was given to how individual teeth are to be marked in a most computationally efficient manner. And with that in mind, techniques searching through three dimensional space seemed like performing a lot of unnecessary computations that would undoubtedly affect the complexity unfavourably. As the three dimensional volume contains information that is redundant for the tooth selection process. The technique developed for approaching this was projecting a threshold version three dimensional volume of interest down to a two dimensional map. This would in essence simplify greatly the calculations required to assess the x- and y-axis borders of the bounding box containing the tooth. And in turn, locating the z-axis boundaries can be performed trivially.

In technical terms, the algorithm would traverse the entire region of interest, marking voxels that have a value higher than that of the threshold, and placing those in a two dimensional map. And that serves as a pre-processing phase in order to allocate teeth positions.

After the projection map is attained, it becomes a much simpler task to mark individual tooth locations. The algorithm developed traverses the two dimensional map looking for *hits*. A *hit* in this particular incident is defined as a pixel that corresponds in location to a voxel that has a value higher than that of the threshold. Once a *hit* is encountered, it is imperative to determine if this in fact was part of a tooth, and if so the size of the tooth. And if it was a tooth, the location should be marked so that it would not be revisited in later iterations of the algorithm. This can be a tricky task as teeth are not usually uniform in shape. Which would make searching for the dimensions of the cube/box that engulfs the tooth give inaccurate results if a simplistic approach was used.

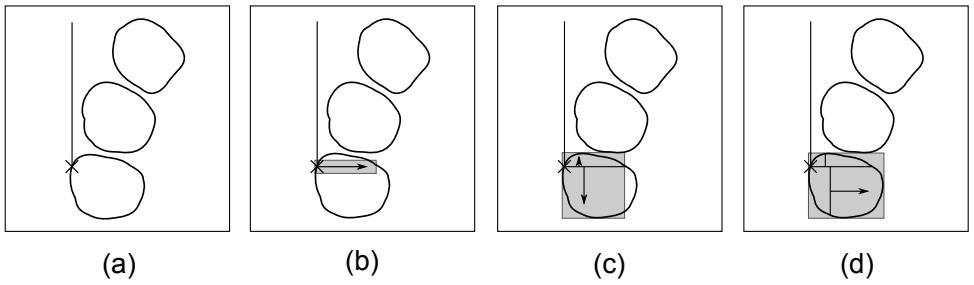

Figure 3.5: *3-Step Depth Search* algorithm, (a) when the algorithm makes a *hit* is starts to assess the size of the high dense area it has encountered, (b) it first steps along the x-axis to determine the end bounds location, (c) along that it inspects the vertical bounds, (d) and finally revisiting to horizontal bounds to make sure the entire area is covered within the bounds.

The technique developed particularly for that search case, which we named *3-Step Depth Search*. In order to accurately find the dimensions, the algorithm which starts from one of the leftmost points, proceeds by expanding its search within three steps. The algorithm first encounters a point of high density from the leftmost point, and as such the lower x-axis bound is set. It then inspects elements to the right of the initial point to see if they lie within

this area of high density. With each step in the x direction, it checks for the upper and lower y limits of the area of high density. And furthermore, it steps again along the x axis looking for the boundary limit. This results in an accurate definition of the bounds of an irregular object.

While this seems like computationally extensive calculations, it is important to point out that is not performed often, and those nested loop searches are likely to end quickly as the area covered by the projection of a tooth onto a two dimensional plane is relatively small. Each iteration halts as soon as it reaches a region outside of what is defined to be a *hit*.

The technique however, as accurate as it may be, does not account for scan imperfections, in which two or more teeth may appear to be connected. For all it takes is one line of pixels connecting one tooth with another for the selection border to engulf both teeth as one. This is handled by a size check just before the tooth's location is about to be stored. If the size is substantially larger than the size of an average tooth, the area is further inspected for possible separations between teeth. After a suitable separation is found, the area is then separated into two teeth. That condition keeps evaluating till the size of the high density area is deemed suitable to be of one tooth.

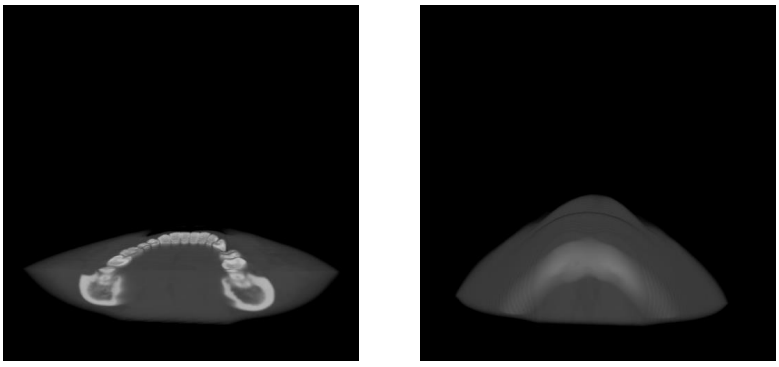

Figure 3.6: Three dimensional reconstruction of a mandible scan showing high density bone formation running through the entire jaw, (a) back view, (b) bottom view.

The methodology discussed above runs for two kinds of scan types, maxilla (upper jaw) and mandible (lower jaw) scans. However, as it turns out,

there was a complication when running with mandible scans. Mandible scans showed a unique bone structure features that are not present for maxilla scans. Those in turn are of very high density in a way that hindered the algorithm useless as the entire jaw structure (with or without presence of teeth) would be marked as one hit spanning all over tooth locations.

Figure 3.6 illustrates the high density bone structure present in mandible scans. This bone formation results in the entire jaw being positively marked as high density tissue that the algorithm uses as a way to allocate teeth. As such, it fails to further mark individual teeth.

The complication was overcome by using statistical methods in order to determine the average positions of teeth in the mandible. And while having the region of interest as the entire jaw, reducing the three dimensional space that gets mapped onto a two dimensional plane for tooth marking in a way that does not let the high density bone structure get in the way of the algorithm running.

The running of the plugin previously discussed results in attaining the input medical scan slices, reconstructed into a three dimensional volume, cropped of unneeded data and with location markers for the positions of existing teeth. While the original purpose of this research was to use those tooth markers to obtain patterns or information that could otherwise assist in tooth implant operation, time allocated for this project and a number of setbacks have disallowed applying further analyses.

3.3 Data mining

The CRISP-DM methodology have been followed. First, the problem understanding phase involved understanding the surgical process and the factors therein. Doing so to be able to find the information that can be integrated and analysed to find the patterns. And following that, a data mining problem definition was obtained. The problem involved finding a predictive model in which an osseointegration measure is to be determined. The main goal being, to find the factor of osseointegration based on the clinical information together with the information obtained from the images.

Then came the data understanding phase, which begins with data collection, further inspection of the dataset to better one's understanding, discovering connections and forming an early hypothesis. Initial raw data that was utilised in order to produce the dataset used for the data mining process was donated by the dentist who put forward the research topic. Originally, data included two separate work sets. One that included 15 attributes about patients. And another that included 15 attributes about implant operations performed on said patients.

The raw data patient attributes contained instances for 32 patients. Those attributes are: patient number, age, sex, consultation frequency, oral hygiene, and boolean attributes for: smoking, bruxism, drugs, diabetes, hypertension, hyperparathyroidism, heart disease and osteoporosis, as well as two string attributes for other conditions and pharmaceutical drugs patients might be on. While the the implant operation data included attributes for 77 implant operations done on the 32 patients listed before. The collected attributes are: patient number, maxillar dental notation, mandibular dental notation, implant type, length, diameter, drilling method, graft, graft material, surgical technique, closing technique, torque, resonance frequency value before osseointegration, osseointegration time, resonance frequency value after osseointegration.

Dental notations use the World Dental Federation notation system (a two-digit numbering standard used to point to a specific tooth in the jaw). The resonance frequency value is a measure of dental stability calculated by a hand-held measuring device allowing clinicians to asses the healing of dental implants. The reading is defined as implant stability quotient which ranges from 0 to 100, 100 being most stable. Readings have a primary value indicating implant stability and a secondary value indicating changes due to bone formation.

Next comes the data preparation phase, which involved cleaning the initial raw data in order to get to a formal dataset to work on for the later phases. Both sets were merged into one, and attributes were assessed to determine their significance for the data mining process. Data cleaning details are further discussed int he following chapter.

After that comes the modelling phase, in which modelling techniques are applied to the dataset. Weka was utilised to apply different classification techniques. Weka provides a wide variety of classification algorithms to be applied to datasets. The dataset was tested with 66 algorithms for the two dental implant stability attributes, *resonance frequency value before osseointegration*, and *resonance frequency value after osseointegration*. Resonance frequency value before osseointegration is the value right after the transplant operation is performed. It is not the ultimate measure of success of an operation, however there is a strong correlation between the resonance frequency value right after the implant operation and after osseointegration. And as such, a predictive classification model was created for the resonance frequency value after the implant, and another for the resonance frequency value after osseointegration using the previous resonance frequency value as an attribute of the dataset model.

Chapter 4

Implementation

This chapter discusses how the approach process discussed in the previous chapter has been implemented.

The application developed for the purpose of this project is developed as *Fiji* plugin. *Fiji* is a cross-platform, open-source package based on *ImageJ*.

4.1 Jaw computed tomography slice analysis

4.1.1 Visualisation

Fiji has a number of plugins that come with the default installation package, as is the case with *ImageJ*. However, *Fiji* is more concerned with plugins that allows for life sciences research[9].

There was a plugin that comes with the default installation that visualises stacks in *DICOM* files, called *3D Viewer*. It however was difficult to interpret for end-users as the visualisation was a single mass blob where tissues exist. There was a necessity of some form of pre-processing. After consideration, it was decided to go with an edge detecting plugin, which too comes with the default installation package. The used plugin is part of the *FeatureJ* package, and is called *FeatureJ Edges*. The plugin runs edge detection algorthims throughout the volume, and the output is a 3D voulme of the dected edges. The result of the use of those two plugins is a diagram easy to interpret and interact with.

After the platform for visualising the stack volume was selected, it was time to assess whether selected platform was suitable for development of the

code that does the tasks required for addressing the problem of this research.

4.1.2 Excess volume cropping

As previously discussed, there was a need to separate the region of interest (jaw region) from the rest of scan volume in order to optimise calculations taking place on the volume through removing unwanted tissue. This takes place over the course of two steps. The first of which is to find the region of highest average density, which corresponds to the mid-skull region that would engulf the least null values when the calculating average cube moves along the scan volume. When that region is found, the application proceeds by removing the volume to the back of the cube. The region almost corresponds to the back-half of the skull, depending on the scanning variables.

After which, the application then proceeds by further removing half of the remaining scan by detecting where the region of interest lies. It looks for the voxel with the highest intensity, which is placed in the remaining half of scan that engulfs the region of interest, and removes the other half from the scan volume. The result of this step is always a 50% reduction. As the volume is divided into two based on the number of slices, and one half of the volume is always removed.

The following is the pseudo-code of the above mentioned functionality.

```
function segment()
    ...
    get sliceWidth;
    get sliceHeight;
    get NumberofSlices;
    set cubeLength, cubeWidth;
    cubeHeight := NumberofSlices;
    HighestAverage := 0;
    HighestAveragePosition := 0;

    // calculating region with highest average density

    for different Y values within image height
        for different voxels within cube with dimensions(
        cubeLength, cubeWidth, cubeHeight)
            get voxelValue;
```

```
    end for

    calculate voxelValueAverage;

    if voxelValueAverage > HighestAverage
        HighestAverage := voxelValueAverage;
        HighestAveragePosition := currentYPosition;
    end if
end for

// removing the back end of the skull

for voxels from HighestAveragePosition till sliceHeight
        voxelValue := 0;
end for

// detecting number of voxels with higher intensity

upperHighIntensity := 0, lowerHighIntensity := 0;

for remaining scan voxels
    get voxelValue;
    if voxelValue > intensityThreshold
        if voxel is in upperHalf of scan
            upperHighIntensity++;
        else
            lowerHighIntensity++;
        end if
    end if
end for

if upperHighIntensity > lowerHighIntensity
    upper := true;
else
    upper := false;
end if

// eliminating the other half of the remaining volume

if upper
    for upper half of remaining scan voxels
```

```
            voxelValue := 0;
        end for
    else
        for lower half of remaining scan voxels
            voxelValue := 0;
        end for
    end if
    ...
end segment
```

4.1.3 Point selection

As discussed in the approach chapter, the point selection part of the code is where most of the plugin functionality is implemented.

The process starts by finding the boundaries of the region of interest, which engulfs all the high dense tissue which at a later point positions of teeth are to be determined. After that, the three dimensional region of interest is projected on a two dimensional boolean plane, where positive values are those that correspond to the voxels that exceed the value of a tooth intensity threshold in the three dimensional volume of interest. Once the projection map is computed, it is looped upon where positive values trigger a searching mechanism named *3-Step Depth Search*, that determines the size of this region of high dense tissue. If the size is big enough for a tooth, the position is allocated for the tooth and marked as such so it does not get included in future iterations of the code. If however the size of the region of interest is bigger than the size of one tooth, it is likely to have been caused by bridging between two or more teeth. The region is inspected for possible separations within the region indicating the beginning or end of one tooth. This carries on until the size of an individual tooth is not computed as too large. In any case, before any tooth is marked, it is checked that the center position of the previous tooth is at an adequate distance from the center position of the current tooth. As it is the case that sometimes teeth are split into two due to the size constraint implemented or scan imperfections. If the distance between the two centres is too small, the two teeth are merged into one tooth engulfing the area that they both are allocated in.

The following is pseudo code that implements the discussed functionality.

```
function teethLocator()
    ...
    get minx, maxx, miny, maxy, minz, maxz;
    //bounding coordinates of the region of interest

    boolean[][] projmap := new boolean[maxx-minx][maxy-miny];

    for voxels within coordinates(minx to maxx, miny to maxy,
    minz to maxz)
        if voxelValue > threshold
            projmap[x][y] := true;
        end if
    end for

    int[][] toothpos := new int[30][4];
    toothc := 0;
    //array and counter for teeth markers

    for points within coordinates(minx to maxx, miny to maxy)

        if projmap[x][y] = true

            for number of marked teeth
                if x and y lie in teethpos
                    break;
                end if
            end for

            //3-step depth search

            offx1:=1, offyp:=1, offyn:=1, offx2:=1;
            //setting up search offsets

            startx:=x, starty:=y, endx:=x, endy:=y;
            //setting up tooth boundaries

            while x+offx1<bound && projmap[x+offxp1][y]=true

                offyp := 1, offyn := 1;

                while y+offyp<bound && projmap[x+offxp1]
```

```
[y+offyp]=true

    offx2 := offx1+1;

    while x+offx2<bound && projmap[x+offx2]
    [y+offyp]=true

        if endx < x+offx2
            endx := x+offx2;
        end if

        offx2++;

    end while

    if endy<y+offyp
        starty := y+offyp;
    end if

    offyp++;

end while

while y-offyn>bound && projmap[x+offxp1]
[y-offyn]=true

    offx2 := offx1+1;

    while x+offx2<bound && projmap[x+offx2]
    [y-offyn]=true

        if endx < x+offx2
            endx := x+offx2;
        end if

        offx2++;

    end while

    if starty>y-offyn
        starty := y-offyn;
```

```
        end if

        offyn++;

    end while

    if endx < x+offx1
        endx := x+offx1;
    end if

end while

//size checking

if toothSize is not too small

    while toothSize is too big
        if endy-starty > endx-startx
            find suitable horizontal separation
            add tooth coordinates to toothpos;
            toothc++;
        else
            find suitable vertical separation
            add tooth coordinates to toothpos;
            toothc++;
        end if

    end while

    if toothc > 0
        if currentToothCenter-previousToothCenter <
        threshold
            merge two teeth;
        end if
    end if

    add tooth coordinates to toothpos
    toothc++;

end if
```

```
        end if
    end for
    ...
end teethLocator
```

4.2 Data mining

Throughout the data preparation phase, attributes were assessed to determine their significance for the data mining process. It was found that the text attributes, "other medical conditions" and "pharmaceutical drugs," will be of little help to the process. And as they were not of great significance, it was decided not to use those. Also, the attributes "implant type" and "drug use" both had only one value each for all the instances. As such, they will not add anything of value to the model, and they were both discarded. As for the "resonance frequency value" attributes, both included the primary and secondary numeric values as strings. But as those values were of high importance, each text value was split into two numeric values, marking the primary and secondary values of the resonance frequency. And finally patient number was discarded after used to merge both data work sets.

The resonance frequency value is the sought variable in the dataset instances. The goal of the data mining process is to define how other factors affect tooth implant stability. As, however, the dataset does not contain a large number of instances, it would become especially difficult to attempt to predict the exact numerical figure of the dental stability quotient. It was decided to discretise the attributes, each around the mean value of its attribute, distinguishing below average instances from above average instances. The resonance frequency right after the implant operation is performed, before osseointegration, has a mean value of 68.58 and a standard of deviation of 8.429, with maximum and minimum values of 86 and 30 respectively. After allowing time for osseointegration, the mean value becomes 72.74 with a 4.601 standard of deviation, 85 as a maximum value and 62 as a minimum value. Consequently, for resonance frequency values before osseointegration, values below 68.58 were replaced with below average, while the remaining values were replaced with above average. As for values after osseointegration, values above 72.74 were replaced with above average while the rest below average.

The final size of the dataset after the data cleaning process was 27 attributes for each of the 77 instances.

For the modelling phase, Weka was then utilised to use classification algorithms which Weka offers to estimate the accuracy of prediction techniques. A total of 66 algorithms compatible with the dataset and the sought attributes. The evaluation fo the results achieved is discussed in the following chapter.

Chapter 5

Results & Discussion

This chapter discusses the result achieved from running the developed Fiji plugin with different test case sets. The later part of the chapter discusses the data mining results.

5.1 Tooth marking

As previously explained the Fiji plugin takes as input medical jaw slices of either the maxilla or the mandible and attempts to locate and mark positions that believes to be of tooth positions. The states that will be used to assess the running of the plugin; *true positive*, in which the plugin correctly identifies and marks a tooth position, *true negative*, correctly not assigning a tooth locator to positions that do not correspond to teeth, *false positive*, incorrectly placing a tooth locator to a position that does not correspond to a tooth, and *false negative*, not placing a tooth locator to a position that corresponds to a tooth.

Throughout the testing period, it was evident that the scan quality has a direct correlation to the quality of results achieved. Other variables kept constant, scans that contained a lot of noise often resulted in poor result quality, where clean scans often result in better result quality.

Figure 5.1 shows a three dimensional reconstruction of the maxilla scan slices. The figure is shown once clear of the tooth markings, and another after a successful run for the test case. The result shows a success, in which there are true positives for all nine teeth positioned in maxilla, along with the absence of false positives.

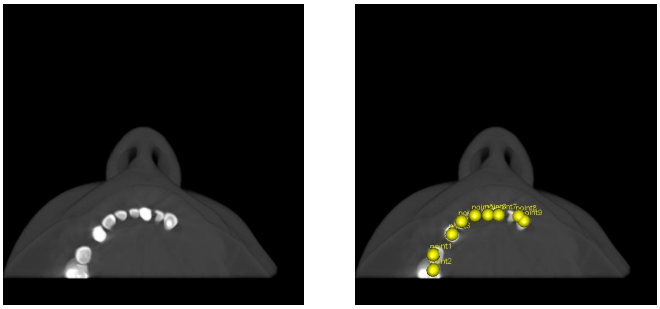

Figure 5.1: Maxilla test case, all teeth are successfully marked with no false positives.

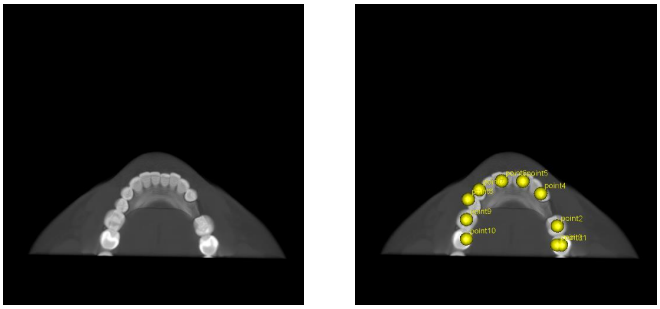

Figure 5.2: Mandible test case, ten of thirteen teeth are successfully marked. The remaining teeth are left unmarked. There are no false positives.

Figure 5.2 shows the plugin run for a mandible test case. The mandible contained thirteen teeth, only ten of them were successfully marked. This means there are three cases of false negatives for that particular case. It is also noticeable that the tooth marker for the second cuspid on the left side of the mandible is not correctly placed at the center of the tooth. The false negatives could be explained by the fact that the incisors and the canine teeth are closely packed. The post processing size segmentation might have not worked for this case as the size of a number of teeth did not reach the segmentation threshold size. This is a difficulty that is hard to get around as teeth sizes range from incisors, which are relatively small in size, to molars, which are quite larger. As for the misplaced marker, after viewing the two dimensional boolean map of the high dense tissue of the volume, it was evident that segmentation between the the second cuspid and first molar has been

performed at an incorrect place, in a way such that part of the molar tooth became attached to the molar. This resulted in expanding the x component of the tooth, and thus centring the marker in the tooth volume of a wrong proportion resulted in a wrong position.

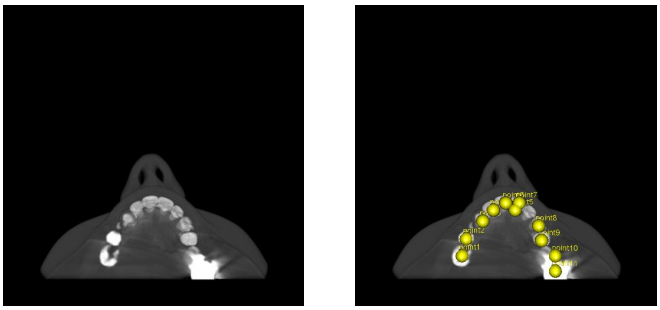

Figure 5.3: Maxilla test case exhibiting little noise, eleven teeth present resulted in eleven markers, one being misplaced. There are no false positives.

Figure 5.3 shows a maxilla test case. The scan however includes some noise as evident in the lower right corner of the volume reconstruction. The plugin run has counted eleven teeth and placed markers for them. However there is one misplaced marker for the lateral incisor tooth. After inspecting the projection boolean map, it was found that further noise not visible in the three dimensional reconstruction has attributed to the wrong placement by enlarging the volume of the high dense tissue to a larger area than the tooth occupies.

Figure 5.4 shows a mandible test case with more noise in the slices to be reconstructed. As a result the quality of the results decreased greatly. Out of the twelve teeth present in the mandible, only six of them were correctly identified and marked. The right second molar has been marked by two markers indicating the plugin code has identified it as two individual teeth. This is likely attributed to the noise of the scan. As the size of molars in previous test cases is similar to the size of that particular molar. The right lateral and central incisors are marked as one tooth. This could have been caused either by the same reason of the test case presented in *figure 5.2* or could have been also due to the noise. For the rest of the bicuspids and molars noise is believed to be the primary cause of erroneous placement of markers or the lack thereof.

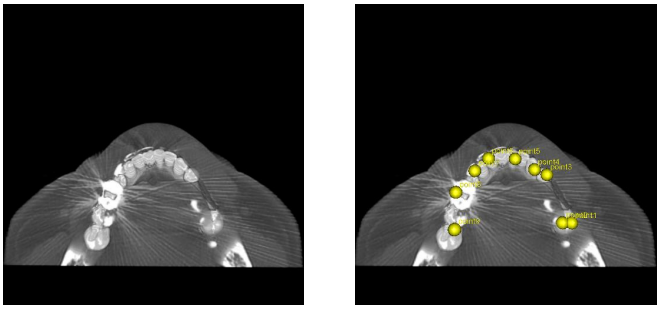

Figure 5.4: Mandible test case, exhibiting more evidence of noise. The jaw contains twelve teeth. Only nine markers were placed, with one tooth marked twice. Eight true positives, one false positive and three false negatives.

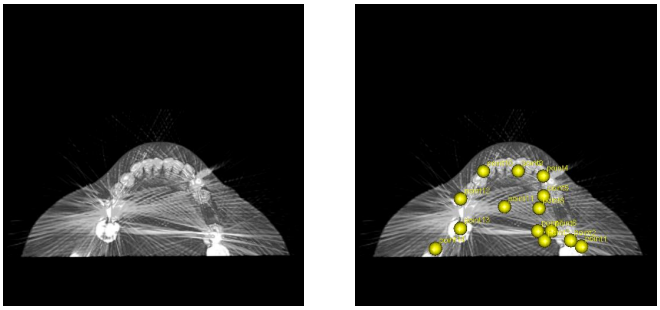

Figure 5.5: Mandible test case, exhibiting much more noise resulting in numerous false positives and false negatives.

Figure 5.5 shows a mandible test case with considerable noise. In fact, information is so distorted one can barely visually identify teeth. However, closer inspection revealed that there are thirteen teeth in the mandible and the plugin run resulted in fourteen markers. Five of those markers are false positives, indicating there were five individual locations where it was computed the presence of enough high dense tissue to declare the position a tooth and place a marker. The remaining makers misplaced for the most part and only for two locations do they mark the center position of a tooth.

The results achieved in cases with high presence of noise is deemed unfit for use. The objective of the plugin is to automate the process of allocating and marking teeth. Cases where the scan slices contain considerable noise

hinder the process by causing false calculations that often result in false positives and false negatives.

5.2 Data mining

While testing the data models that predict the stability right after the implant operation, before osseointegration, data regarding stability after osseointegration was eliminated. As the idea is to create a predictive model that relies on information available at the time. Testing the stability value after osseointegration, included the stability parameter before osseointegration. And thus resulted in an overall better classification success rate due to the likely strong correlation between the stability values.

For the before osseointegration stability value testing, different algorithms tested on the dataset resulted in classification success that ranged from 46.75% to 76.62%. Of the better performing models, Naïve Bayes (71.43%), Lazy Bagging (71.43%), Classification Via Regression (71.43%), LogitBoost (76.62%), Random Committee (75.32%), Rotation Forest (70.13%), Voting Feature Intervals (72.73%), Decision Table/Naïve Bayes Hybrid (71.43%), ADTree (74.03%), Functional Tree (71.43%), NB-Tree (74.02%) and Random Forest (70.13%).

As for the after osseointegration parameter, different models have a correct classification rate of 62.34% to 84.42%. The models differ with the use of the stability value before osseointegration as this offer a closer indication to the stability factor after. The successful models in this case are, Bayes Net (83.12%), Naïve Bayes (84.42%), Sequential Minimal Optimisation (80.52%), Attribute Selected Classifier (80.52%), Bagging (83.12%), Dagging (81.82%), Filtered Classifier (80.52%), Random Sub-Space (81.82%), Rotation Forest (81.82%), Voting Feature Intervals (81.82%), OneR (83.12%), J48 Graft (80.52%), Logistic Model Trees (80.52%), Random Forest (81.82%) and Random Tree (80.52%).

The inclusiveness of the outcome of classification models could be attributed to two factors. The first being there is no strong direct correlation between the attributes given in the dataset and the stability parameters. And the second, the number of instances in the dataset is relatively small for classification techniques to operate adequately.

The appendix at the end of this document contains a detailed list of the summary of results achieved for the different classification and regression algorithms tested.

Of the well scoring models tested, it was decided the Naïve Bayes model provides a fairly good prediction model for both cases, before as well as after osseointegration.

5.2.1 Naïve Bayes classification

Naïve Bayes is one of the most powerful inductive learning algorithms for machine learning. Classification is done by applying Bayes rule to compute the portability of the class attribute given a set of attributes. It is a simple form of Bayesian networking in which all attributes are assumed to be independent of the class variable, which is called conditional independence. The conditional independence might not be applicable in real world examples, however the classifier still produce favourable results. This is likely due to the fact that the classification results most depend on dependencies among all attributes. It is that dependence between the attributes that affects the classification[14][15].

Chapter 6

Conclusion & Future Work

6.1 Conclusion

The application developed for the purpose of this research performs computations to automatically discover and mark teeth positions in medical scans. It does so by firstly by performing analysis of different scan parts and choosing which parts to discard and which to attain for further processing. After that, more fine grain processing is performed, in which structures of high density are examined and checked with size constraints. Once the structure is determined is big enough to be that of a tooth, one further size constraint is checked in order to establish that the detected tooth is too big and was the likely result of scan imperfections merging two or more teeth together. If that was the case, computations are performed to figure out likely tooth boundaries within the high density mass. After all those checks have been performed, and the tooth's position is ready to be marked, one final check is performed. The last constraint calculates if the current tooth's center point is within close proximity with the center point of the tooth before. If the test is positive, it means this is likely caused due to splitting one tooth onto two volumes. Both teeth are merged back together to be covered by one marker.

The overall accuracy of the markers depends largely on the quality of the scan files. Scan files with considerable noise result in poor quality of results. It is thus imperative to consider the quality in terms of input to obtain quality output.

As for the data mining approach, two probabilistic classifier prediction models based on Bayes' theorem were obtained. The two models attempt to forecast the stability measure of a tooth implant based on other information

acquired, one before the operation is performed, and one right after it is performed and the initial stability value is attained. The model success rate of initial classification ranges from 71.43% to 84.42%.

6.2 Future work

As the time allocated for the project disallowed analysis beyond marking positions of individual teeth within scans of maxillae and mandibles, it therefore leaves an opportunity for further analysis based on this work and findings. Perhaps even beyond the application for dental implants. It is possible to use the application developed, or the idea behind the implementation, for a more general approach of other dental problems. And apart from dental problems, the implementation could be adjusted in order to be used for various medical problems. Automated way of detecting volumes based on density (high or low) in various body parts for different applications. And lastly one further room for improvement here might be introduction of additional algorithm constraints to help increase the precision of the resultant markers.

It remains to be seen if the data mining models could be improved with the introduction of more information to the dataset. It is likely that increasing the dataset instances results in better pattern matching and thus better classification. Additional attributes that provide the classification and regression algorithms more information to work on might result in improvement in classification success rate of the models.

And finally it would be beneficial if the two implementations were fully integrated into one application. Doing so was the motivation of this study. Knowledge discovery and pattern detection working along side tissue density analysis, is likely to improve the results achieved for both processes.

Appendix.
Data mining results using different classifiers

Resonance frequency value before osseointegration

Bayes:

Bayes Net:

Correctly Classified Instances	52	67.5325%
Incorrectly Classified Instances	25	32.4675%
Kappa statistic	0.3446	
Mean absolute error	0.3297	
Root mean squared error	0.4806	
Relative absolute error	67.1864%	
Root relative squared error	96.9831%	
Total Number of Instances	77	

Naïve Bayes:

Correctly Classified Instances	55	71.4286%
Incorrectly Classified Instances	22	28.5714%
Kappa statistic	0.4296	
Mean absolute error	0.3069	
Root mean squared error	0.4625	
Relative absolute error	62.5357%	
Root relative squared error	93.3351%	
Total Number of Instances	77	

Naïve Bayes Simple:

Correctly Classified Instances	55	71.4286%
Incorrectly Classified Instances	22	28.5714%
Kappa statistic	0.4296	
Mean absolute error	0.3015	
Root mean squared error	0.4578	
Relative absolute error	61.4379%	
Root relative squared error	92.3787%	
Total Number of Instances	77	

Naïve Bayes Updatable:

Correctly Classified Instances	55	71.4286%
Incorrectly Classified Instances	22	28.5714%
Kappa statistic	0.4296	
Mean absolute error	0.3069	
Root mean squared error	0.4625	
Relative absolute error	62.5357%	
Root relative squared error	93.3351%	
Total Number of Instances	77	

Functions:

Logistic:

Correctly Classified Instances	50	64.9351%
Incorrectly Classified Instances	27	35.0649%
Kappa statistic	0.2921	
Mean absolute error	0.3612	
Root mean squared error	0.5764	
Relative absolute error	73.6136%	
Root relative squared error	116.319%	
Total Number of Instances	77	

Multi-player Perception:

Correctly Classified Instances	49	63.6364%
Incorrectly Classified Instances	28	36.3636%
Kappa statistic	0.2519	
Mean absolute error	0.3576	
Root mean squared error	0.5635	
Relative absolute error	72.871%	
Root relative squared error	113.716%	
Total Number of Instances	77	

RBF Network:

Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3588	
Mean absolute error	0.3656	
Root mean squared error	0.4834	
Relative absolute error	74.509%	

Root relative squared error	97.5502%	
Total Number of Instances	77	

Simple Logistic:

Correctly Classified Instances	44	57.1429%
Incorrectly Classified Instances	33	42.8571%
Kappa statistic	0.1283	
Mean absolute error	0.4237	
Root mean squared error	0.5027	
Relative absolute error	86.3485%	
Root relative squared error	101.449%	
Total Number of Instances	77	

SMO:

Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3731	
Mean absolute error	0.3117	
Root mean squared error	0.5583	
Relative absolute error	63.5179%	
Root relative squared error	112.662%	
Total Number of Instances	77	

S Pegasos:

Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0.2509	
Mean absolute error	0.3766	
Root mean squared error	0.6137	
Relative absolute error	76.7508%	
Root relative squared error	123.842%	
Total Number of Instances	77	

Voted Perceptron:

Correctly Classified Instances	50	64.9351%
Incorrectly Classified Instances	27	35.0649%
Kappa statistic	0.2158	
Mean absolute error	0.3512	
Root mean squared error	0.5817	
Relative absolute error	71.5762%	
Root relative squared error	117.389%	
Total Number of Instances	77	

Lazy:

IB1:

Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0.2222	
Mean absolute error	0.3766	
Root mean squared error	0.6137	
Relative absolute error	76.7508%	
Root relative squared error	123.842%	
Total Number of Instances	77	

IBk:

Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0.2222	
Mean absolute error	0.3801	
Root mean squared error	0.6052	
Relative absolute error	77.4575%	
Root relative squared error	122.126%	
Total Number of Instances	77	

K-Star:

Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3488	
Mean absolute error	0.346	
Root mean squared error	0.552	
Relative absolute error	70.502%	
Root relative squared error	111.385%	
Total Number of Instances	77	

LWL:

Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3588	
Mean absolute error	0.3559	
Root mean squared error	0.4482	
Relative absolute error	72.5323%	
Root relative squared error	90.4493%	
Total Number of Instances	77	

Meta:

Ada Boost M1:

Correctly Classified Instances	51	66.2338%
Incorrectly Classified Instances	26	33.7662%
Kappa statistic	0.2946	
Mean absolute error	0.3561	
Root mean squared error	0.4668	
Relative absolute error	72.5626%	
Root relative squared error	94.1942%	
Total Number of Instances	77	

Attribute Selected Classifier:

Correctly Classified Instances	47	61.039%
Incorrectly Classified Instances	30	38.961%
Kappa statistic	0.186	
Mean absolute error	0.4138	
Root mean squared error	0.5187	
Relative absolute error	84.3354%	
Root relative squared error	104.665%	
Total Number of Instances	77	

Bagging:

Correctly Classified Instances	55	71.4286%
Incorrectly Classified Instances	22	28.5714%
Kappa statistic	0.4254	
Mean absolute error	0.3585	
Root mean squared error	0.4394	
Relative absolute error	73.059%	
Root relative squared error	88.6705%	
Total Number of Instances	77	

Classification Via Clustering:

Correctly Classified Instances	45	58.4416%
Incorrectly Classified Instances	32	41.5584%
Kappa statistic	0.1884	
Mean absolute error	0.4156	

Root mean squared error	0.6447	
Relative absolute error	84.6905%	
Root relative squared error	130.090%	
Total Number of Instances	77	
Classification Via Regression:		
Correctly Classified Instances	55	71.4286%
Incorrectly Classified Instances	22	28.5714%
Kappa statistic	0.4167	
Mean absolute error	0.3812	
Root mean squared error	0.4657	
Relative absolute error	77.6936%	
Root relative squared error	93.9808%	
Total Number of Instances	77	
CV Parameter Selection:		
Correctly Classified Instances	44	57.1429%
Incorrectly Classified Instances	33	42.8571%
Kappa statistic	0	
Mean absolute error	0.4907	
Root mean squared error	0.4955	
Relative absolute error	100%	
Root relative squared error	100%	
Total Number of Instances	77	
Dagging:		
Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0.1976	
Mean absolute error	0.413	
Root mean squared error	0.4726	
Relative absolute error	84.1612%	
Root relative squared error	95.3753%	
Total Number of Instances	77	
END:		
Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3731	
Mean absolute error	0.3384	
Root mean squared error	0.4985	
Relative absolute error	68.9702%	
Root relative squared error	100.602%	
Total Number of Instances	77	
Filtered Classifier:		
Correctly Classified Instances	51	66.2338%
Incorrectly Classified Instances	26	33.7662%
Kappa statistic	0.3106	
Mean absolute error	0.3888	
Root mean squared error	0.4979	
Relative absolute error	79.2243%	
Root relative squared error	100.475%	
Total Number of Instances	77	
Grading:		
Correctly Classified Instances	44	57.1429%
Incorrectly Classified Instances	33	42.8571%
Kappa statistic	0	
Mean absolute error	0.4286	
Root mean squared error	0.6547	
Relative absolute error	87.3371%	
Root relative squared error	132.107%	
Total Number of Instances	77	
LogitBoost:		
Correctly Classified Instances	59	76.6234%
Incorrectly Classified Instances	18	23.3766%
Kappa statistic	0.5227	
Mean absolute error	0.304	
Root mean squared error	0.4228	
Relative absolute error	61.9565%	
Root relative squared error	85.3216%	
Total Number of Instances	77	
Multi Boost AB:		
Correctly Classified Instances	49	63.6364%
Incorrectly Classified Instances	28	36.3636%
Kappa statistic	0.2462	
Mean absolute error	0.3522	
Root mean squared error	0.5386	
Relative absolute error	71.7689%	
Root relative squared error	108.683%	
Total Number of Instances	77	
Multi Class Classifier:		
Correctly Classified Instances	50	64.9351%
Incorrectly Classified Instances	27	35.0649%
Kappa statistic	0.2921	
Mean absolute error	0.3612	
Root mean squared error	0.5764	
Relative absolute error	73.6136%	
Root relative squared error	116.316%	
Total Number of Instances	77	
Multi Scheme:		
Correctly Classified Instances	44	57.1429%
Incorrectly Classified Instances	33	42.8571%
Kappa statistic	0	
Mean absolute error	0.4907	
Root mean squared error	0.4955	
Relative absolute error	100%	
Root relative squared error	100%	
Total Number of Instances	77	
Nested Dichotomies, Class Balanced ND:		
Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3731	
Mean absolute error	0.3384	
Root mean squared error	0.4985	
Relative absolute error	68.9702%	
Root relative squared error	100.602%	
Total Number of Instances	77	
Nested Dichotomies, Data Near Balanced ND:		
Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3731	
Mean absolute error	0.3384	
Root mean squared error	0.4985	
Relative absolute error	68.9702%	
Root relative squared error	100.602%	
Total Number of Instances	77	

Nested Dichotomies, ND:		
Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3731	
Mean absolute error	0.3384	
Root mean squared error	0.4985	
Relative absolute error	68.9702%	
Root relative squared error	100.602%	
Total Number of Instances	77	

Ordinal Class Classifier:		
Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3731	
Mean absolute error	0.3384	
Root mean squared error	0.4985	
Relative absolute error	68.9702%	
Root relative squared error	100.602%	
Total Number of Instances	77	

Raced Incremental LogitBoost:		
Correctly Classified Instances	44	57.1429%
Incorrectly Classified Instances	33	42.8571%
Kappa statistic	0	
Mean absolute error	0.4907	
Root mean squared error	0.4955	
Relative absolute error	100%	
Root relative squared error	100%	
Total Number of Instances	77	

Random Committee:		
Correctly Classified Instances	58	75.3247%
Incorrectly Classified Instances	19	24.6753%
Kappa statistic	0.5019	
Mean absolute error	0.3095	
Root mean squared error	0.4456	
Relative absolute error	63.066%	
Root relative squared error	89.9243%	
Total Number of Instances	77	

Random Sub Space:		
Correctly Classified Instances	50	64.9351%
Incorrectly Classified Instances	27	35.0649%
Kappa statistic	0.2703	
Mean absolute error	0.3949	
Root mean squared error	0.4472	
Relative absolute error	80.4834%	
Root relative squared error	90.2376%	
Total Number of Instances	77	

Rotation Forest:		
Correctly Classified Instances	54	70.1299%
Incorrectly Classified Instances	23	29.8701%
Kappa statistic	0.3878	
Mean absolute error	0.3438	
Root mean squared error	0.4511	
Relative absolute error	70.0655%	
Root relative squared error	91.041%	
Total Number of Instances	77	

Stacking:

Correctly Classified Instances	44	57.1429%
Incorrectly Classified Instances	33	42.8571%
Kappa statistic	0	
Mean absolute error	0.4907	
Root mean squared error	0.4955	
Relative absolute error	100%	
Root relative squared error	100%	
Total Number of Instances	77	

Stacking C:

Correctly Classified Instances	44	57.1429%
Incorrectly Classified Instances	33	42.8571%
Kappa statistic	0	
Mean absolute error	0.4908	
Root mean squared error	0.4957	
Relative absolute error	100.011%	
Root relative squared error	100.023%	
Total Number of Instances	77	

Threshold Selector:

Correctly Classified Instances	36	46.7532%
Incorrectly Classified Instances	41	53.2468%
Kappa statistic	0.0205	
Mean absolute error	0.4469	
Root mean squared error	0.5857	
Relative absolute error	91.0667%	
Root relative squared error	118.196%	
Total Number of Instances	77	

Vote:

Correctly Classified Instances	44	57.1429%
Incorrectly Classified Instances	33	42.8571%
Kappa statistic	0	
Mean absolute error	0.4907	
Root mean squared error	0.4955	
Relative absolute error	100%	
Root relative squared error	100%	
Total Number of Instances	77	

Misc:

Hyper Pipes:		
Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0.1646	
Mean absolute error	0.4954	
Root mean squared error	0.4955	
Relative absolute error	100.951%	
Root relative squared error	99.9862%	
Total Number of Instances	77	

VFI:		
Correctly Classified Instances	56	72.7273%
Incorrectly Classified Instances	21	27.2727%
Kappa statistic	0.4535	
Mean absolute error	0.381	
Root mean squared error	0.4751	
Relative absolute error	77.6338%	
Root relative squared error	95.8743%	
Total Number of Instances	77	

Rules:

Conjunctive Rule:

Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0.1781	
Mean absolute error	0.4394	
Root mean squared error	0.4869	
Relative absolute error	89.5349%	
Root relative squared error	98.2536%	
Total Number of Instances	77	

Decision Table:

Correctly Classified Instances	47	61.039%
Incorrectly Classified Instances	30	38.961%
Kappa statistic	0.1667	
Mean absolute error	0.4482	
Root mean squared error	0.4996	
Relative absolute error	91.333%	
Root relative squared error	100.828%	
Total Number of Instances	77	

DTNB:

Correctly Classified Instances	55	71.4286%
Incorrectly Classified Instances	22	28.5714%
Kappa statistic	0.4254	
Mean absolute error	0.3326	
Root mean squared error	0.4299	
Relative absolute error	67.775%	
Root relative squared error	86.747%	
Total Number of Instances	77	

JRip:

Correctly Classified Instances	50	64.9351%
Incorrectly Classified Instances	27	35.0649%
Kappa statistic	0.2814	
Mean absolute error	0.4045	
Root mean squared error	0.5206	
Relative absolute error	82.4293%	
Root relative squared error	105.055%	
Total Number of Instances	77	

NNge:

Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3538	
Mean absolute error	0.3117	
Root mean squared error	0.5583	
Relative absolute error	63.5179%	
Root relative squared error	112.662%	
Total Number of Instances	77	

OneR:

Correctly Classified Instances	43	55.8442%
Incorrectly Classified Instances	34	44.1558%
Kappa statistic	0.1185	
Mean absolute error	0.4416	
Root mean squared error	0.6645	
Relative absolute error	89.9837%	
Root relative squared error	134.094%	
Total Number of Instances	77	

PART:

Correctly Classified Instances	50	64.9351%
Incorrectly Classified Instances	27	35.0649%
Kappa statistic	0.2921	
Mean absolute error	0.3701	
Root mean squared error	0.5382	
Relative absolute error	75.4255%	
Root relative squared error	108.610%	
Total Number of Instances	77	

Ridor:

Correctly Classified Instances	49	63.6364%
Incorrectly Classified Instances	28	36.3636%
Kappa statistic	0.2462	
Mean absolute error	0.3636	
Root mean squared error	0.603	
Relative absolute error	74.1042%	
Root relative squared error	121.688%	
Total Number of Instances	77	

Zero R:

Correctly Classified Instances	44	57.1429%
Incorrectly Classified Instances	33	42.8571%
Kappa statistic	0	
Mean absolute error	0.4907	
Root mean squared error	0.4955	
Relative absolute error	100%	
Root relative squared error	100%	
Total Number of Instances	77	

Trees:

AD Tree:

Correctly Classified Instances	57	74.026%
Incorrectly Classified Instances	20	25.974%
Kappa statistic	0.4697	
Mean absolute error	0.3593	
Root mean squared error	0.4483	
Relative absolute error	73.2143%	
Root relative squared error	90.4604%	
Total Number of Instances	77	

BF Tree:

Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3636	
Mean absolute error	0.3662	
Root mean squared error	0.4773	
Relative absolute error	74.6199%	
Root relative squared error	96.3261%	
Total Number of Instances	77	

Decision Stump:

Correctly Classified Instances	45	58.4416%
Incorrectly Classified Instances	32	41.5584%
Kappa statistic	0.1515	
Mean absolute error	0.454	
Root mean squared error	0.4985	
Relative absolute error	92.5101%	
Root relative squared error	100.6039%	

Total Number of Instances	77	

FT:

Correctly Classified Instances	55	71.4286%
Incorrectly Classified Instances	22	28.5714%
Kappa statistic	0.4211	
Mean absolute error	0.3375	
Root mean squared error	0.5045	
Relative absolute error	68.7876%	
Root relative squared error	101.803%	
Total Number of Instances	77	

J48:

Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3731	
Mean absolute error	0.3384	
Root mean squared error	0.4985	
Relative absolute error	68.9702%	
Root relative squared error	100.602%	
Total Number of Instances	77	

J48 Graft:

Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3731	
Mean absolute error	0.3384	
Root mean squared error	0.4985	
Relative absolute error	68.9702%	
Root relative squared error	100.602%	
Total Number of Instances	77	

LAD Tree:

Correctly Classified Instances	52	67.5325%
Incorrectly Classified Instances	25	32.4675%
Kappa statistic	0.3346	
Mean absolute error	0.3544	
Root mean squared error	0.5444	
Relative absolute error	72.2255%	
Root relative squared error	109.859%	
Total Number of Instances	77	

LMT:

Correctly Classified Instances	49	63.6364%
Incorrectly Classified Instances	28	36.3636%
Kappa statistic	0.2632	
Mean absolute error	0.3893	
Root mean squared error	0.4999	
Relative absolute error	79.3442%	
Root relative squared error	100.875%	
Total Number of Instances	77	

NB Tree:

Correctly Classified Instances	57	74.026%
Incorrectly Classified Instances	20	25.974%
Kappa statistic	0.4737	
Mean absolute error	0.2866	
Root mean squared error	0.4672	
Relative absolute error	58.3952%	
Root relative squared error	94.2863%	
Total Number of Instances	77	

Random Forest:

Correctly Classified Instances	54	70.1299%
Incorrectly Classified Instances	23	29.8701%
Kappa statistic	0.3831	
Mean absolute error	0.3285	
Root mean squared error	0.431	
Relative absolute error	66.941%	
Root relative squared error	86.9828%	
Total Number of Instances	77	

Random Tree:

Correctly Classified Instances	51	66.2338%
Incorrectly Classified Instances	26	33.7662%
Kappa statistic	0.3106	
Mean absolute error	0.3485	
Root mean squared error	0.5745	
Relative absolute error	71.0161%	
Root relative squared error	115.936%	
Total Number of Instances	77	

REP Tree:

Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3636	
Mean absolute error	0.3377	
Root mean squared error	0.4404	
Relative absolute error	68.8274%	
Root relative squared error	88.876%	
Total Number of Instances	77	

Simple Cart:

Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3636	
Mean absolute error	0.3503	
Root mean squared error	0.471	
Relative absolute error	71.3869%	
Root relative squared error	95.0404%	
Total Number of Instances	77	

User Classifier:

Correctly Classified Instances	44	57.1429%
Incorrectly Classified Instances	33	42.8571%
Kappa statistic	0	
Mean absolute error	0.4904	
Root mean squared error	0.4956	
Relative absolute error	99.9454%	
Root relative squared error	100.003%	
Total Number of Instances	77	

Resonance frequency value after osseointegration

Bayers:		
Bayes Net:		
Correctly Classified Instances	64	83.1169%
Incorrectly Classified Instances	13	16.8831%
Kappa statistic	0.638	
Mean absolute error	0.1883	
Root mean squared error	0.3768	
Relative absolute error	40.0104%	
Root relative squared error	77.7247%	
Total Number of Instances	77	
Naïve Bayes:		
Correctly Classified Instances	65	84.4156%
Incorrectly Classified Instances	12	15.5844%
Kappa statistic	0.6726	
Mean absolute error	0.172	
Root mean squared error	0.3636	
Relative absolute error	36.541%	
Root relative squared error	74.9882%	
Total Number of Instances	77	
Naïve Bayes Simple:		
Correctly Classified Instances	65	84.4156%
Incorrectly Classified Instances	12	15.5844%
Kappa statistic	0.6726	
Mean absolute error	0.1761	
Root mean squared error	0.3683	
Relative absolute error	37.4187%	
Root relative squared error	75.9631%	
Total Number of Instances	77	
Naïve Bayes Updatable:		
Correctly Classified Instances	65	84.4156%
Incorrectly Classified Instances	12	15.5844%
Kappa statistic	0.6726	
Mean absolute error	0.172	
Root mean squared error	0.3636	
Relative absolute error	36.541%	
Root relative squared error	74.9882%	
Total Number of Instances	77	
Functions:		
Logistic:		
Correctly Classified Instances	54	70.1299%
Incorrectly Classified Instances	23	29.8701%
Kappa statistic	0.3506	
Mean absolute error	0.2986	
Root mean squared error	0.5464	
Relative absolute error	63.4485%	
Root relative squared error	112.700%	
Total Number of Instances	77	
Multilayer Perception:		
Correctly Classified Instances	59	76.6234%
Incorrectly Classified Instances	18	23.3766%
Kappa statistic	0.4953	
Mean absolute error	0.2305	
Root mean squared error	0.448	
Relative absolute error	48.9731%	
Root relative squared error	92.4111%	
Total Number of Instances	77	
RBF Network:		
Correctly Classified Instances	61	79.2208%
Incorrectly Classified Instances	16	20.7792%
Kappa statistic	0.5692	
Mean absolute error	0.23	
Root mean squared error	0.3935	
Relative absolute error	48.8627%	
Root relative squared error	81.1579%	
Total Number of Instances	77	
Simple Logistic:		
Correctly Classified Instances	60	77.9221%
Incorrectly Classified Instances	17	22.0779%
Kappa statistic	0.52	
Mean absolute error	0.2973	
Root mean squared error	0.3984	
Relative absolute error	63.1583%	
Root relative squared error	82.1693%	
Total Number of Instances	77	
SMO:		
Correctly Classified Instances	62	80.5195%
Incorrectly Classified Instances	15	19.4805%
Kappa statistic	0.5823	
Mean absolute error	0.1948	
Root mean squared error	0.4414	
Relative absolute error	41.39%	
Root relative squared error	91.0386%	
Total Number of Instances	77	
S Pegasos:		
Correctly Classified Instances	55	71.4286%
Incorrectly Classified Instances	22	28.5714%
Kappa statistic	0.3831	
Mean absolute error	0.2857	
Root mean squared error	0.5345	
Relative absolute error	60.7054%	
Root relative squared error	110.253%	
Total Number of Instances	77	
Voted Perceptron:		
Correctly Classified Instances	50	64.9351%
Incorrectly Classified Instances	27	35.0649%
Kappa statistic	0.0996	
Mean absolute error	0.3485	
Root mean squared error	0.5798	
Relative absolute error	74.0458%	
Root relative squared error	119.599%	
Total Number of Instances	77	

Root mean squared error	0.4172	
Relative absolute error	54.7281%	
Root relative squared error	86.0595%	
Total Number of Instances	77	

Lazy:

IB1:

Correctly Classified Instances	54	70.1299%
Incorrectly Classified Instances	23	29.8701%
Kappa statistic	0.3595	
Mean absolute error	0.2987	
Root mean squared error	0.5465	
Relative absolute error	63.4647%	
Root relative squared error	112.731%	
Total Number of Instances	77	

IBk:

Correctly Classified Instances	54	70.1299%
Incorrectly Classified Instances	23	29.8701%
Kappa statistic	0.3595	
Mean absolute error	0.3044	
Root mean squared error	0.539	
Relative absolute error	64.6657%	
Root relative squared error	111.176%	
Total Number of Instances	77	

K-Star:

Correctly Classified Instances	52	67.5325%
Incorrectly Classified Instances	25	32.4675%
Kappa statistic	0.3132	
Mean absolute error	0.3251	
Root mean squared error	0.551	
Relative absolute error	69.0763%	
Root relative squared error	113.653%	
Total Number of Instances	77	

LWL:

Correctly Classified Instances	59	76.6234%
Incorrectly Classified Instances	18	23.3766%
Kappa statistic	0.4953	
Mean absolute error	0.2852	
Root mean squared error	0.4213	
Relative absolute error	60.5983%	
Root relative squared error	86.8995%	
Total Number of Instances	77	

Meta:

Ada Boost M1:

Correctly Classified Instances	58	75.3247%
Incorrectly Classified Instances	19	24.6753%
Kappa statistic	0.4781	
Mean absolute error	0.2607	
Root mean squared error	0.4185	
Relative absolute error	55.3929%	
Root relative squared error	86.3116%	
Total Number of Instances	77	

Attribute Selected Classifier:

Correctly Classified Instances	62	80.5195%
Incorrectly Classified Instances	15	19.4805%
Kappa statistic	0.5765	
Mean absolute error	0.2576	

Bagging:

Correctly Classified Instances	64	83.1169%
Incorrectly Classified Instances	13	16.8831%
Kappa statistic	0.6277	
Mean absolute error	0.2814	
Root mean squared error	0.3743	
Relative absolute error	59.7943%	
Root relative squared error	77.2016%	
Total Number of Instances	77	

Classification via Clustering:

Correctly Classified Instances	51	66.2338%
Incorrectly Classified Instances	26	33.7662%
Kappa statistic	0.3268	
Mean absolute error	0.3377	
Root mean squared error	0.5811	
Relative absolute error	71.7427%	
Root relative squared error	119.857%	
Total Number of Instances	77	

Classification via Regression:

Correctly Classified Instances	60	77.9221%
Incorrectly Classified Instances	17	22.0779%
Kappa statistic	0.5392	
Mean absolute error	0.2465	
Root mean squared error	0.3543	
Relative absolute error	52.3799%	
Root relative squared error	73.0873%	
Total Number of Instances	77	

CV Parameter Selection:

Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0	
Mean absolute error	0.4707	
Root mean squared error	0.4848	
Relative absolute error	100%	
Root relative squared error	100%	
Total Number of Instances	77	

Dagging:

Correctly Classified Instances	63	81.8182%
Incorrectly Classified Instances	14	18.1818%
Kappa statistic	0.6074	
Mean absolute error	0.3	
Root mean squared error	0.3798	
Relative absolute error	63.7406%	
Root relative squared error	78.3496%	
Total Number of Instances	77	

END:

Correctly Classified Instances	60	77.9221%
Incorrectly Classified Instances	17	22.0779%
Kappa statistic	0.5132	
Mean absolute error	0.2614	
Root mean squared error	0.4241	
Relative absolute error	55.5291%	

Root relative squared error	87.4735%	
Total Number of Instances	77	
Filtered Classifier:		
Correctly Classified Instances	62	80.5195%
Incorrectly Classified Instances	15	19.4805%
Kappa statistic	0.5765	
Mean absolute error	0.2573	
Root mean squared error	0.3985	
Relative absolute error	54.6747%	
Root relative squared error	82.2002%	
Total Number of Instances	77	
Grading:		
Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0	
Mean absolute error	0.3766	
Root mean squared error	0.6137	
Relative absolute error	80.0207%	
Root relative squared error	126.584%	
Total Number of Instances	77	
LogitBoost:		
Correctly Classified Instances	59	76.6234%
Incorrectly Classified Instances	18	23.3766%
Kappa statistic	0.5022	
Mean absolute error	0.2495	
Root mean squared error	0.3963	
Relative absolute error	53.0016%	
Root relative squared error	81.7522%	
Total Number of Instances	77	
Multi-Boost AB:		
Correctly Classified Instances	61	79.2208%
Incorrectly Classified Instances	16	20.7792%
Kappa statistic	0.5513	
Mean absolute error	0.218	
Root mean squared error	0.4514	
Relative absolute error	46.3175%	
Root relative squared error	93.1051%	
Total Number of Instances	77	
Multi-Class Classifier:		
Correctly Classified Instances	54	70.1299%
Incorrectly Classified Instances	23	29.8701%
Kappa statistic	0.3506	
Mean absolute error	0.2986	
Root mean squared error	0.5464	
Relative absolute error	63.4485%	
Root relative squared error	112.700%	
Total Number of Instances	77	
Multi-Scheme:		
Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0	
Mean absolute error	0.4707	
Root mean squared error	0.4848	
Relative absolute error	100%	
Root relative squared error	100%	
Total Number of Instances	77	

Nested Dichotomies, Class Balanced ND:		
Correctly Classified Instances	60	77.9221%
Incorrectly Classified Instances	17	22.0779%
Kappa statistic	0.5132	
Mean absolute error	0.2614	
Root mean squared error	0.4241	
Relative absolute error	55.5291%	
Root relative squared error	87.4735%	
Total Number of Instances	77	
Nested Dichotomies, Data Near Balanced ND:		
Correctly Classified Instances	60	77.9221%
Incorrectly Classified Instances	17	22.0779%
Kappa statistic	0.5132	
Mean absolute error	0.2614	
Root mean squared error	0.4241	
Relative absolute error	55.5291%	
Root relative squared error	87.4735%	
Total Number of Instances	77	
Nested Dichotomies, ND:		
Correctly Classified Instances	60	77.9221%
Incorrectly Classified Instances	17	22.0779%
Kappa statistic	0.5132	
Mean absolute error	0.2614	
Root mean squared error	0.4241	
Relative absolute error	55.5291%	
Root relative squared error	87.4735%	
Total Number of Instances	77	
Ordinal Class Classifier:		
Correctly Classified Instances	60	77.9221%
Incorrectly Classified Instances	17	22.0779%
Kappa statistic	0.5132	
Mean absolute error	0.2614	
Root mean squared error	0.4241	
Relative absolute error	55.5291%	
Root relative squared error	87.4735%	
Total Number of Instances	77	
Raced Incremental LogitBoost:		
Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0	
Mean absolute error	0.4707	
Root mean squared error	0.4848	
Relative absolute error	100%	
Root relative squared error	100%	
Total Number of Instances	77	
Random Committee:		
Correctly Classified Instances	59	76.6234%
Incorrectly Classified Instances	18	23.3766%
Kappa statistic	0.4953	
Mean absolute error	0.2549	
Root mean squared error	0.3873	
Relative absolute error	54.1562%	
Root relative squared error	79.8814%	
Total Number of Instances	77	
Random Sub Space:		

Correctly Classified Instances	63	81.8182%
Incorrectly Classified Instances	14	18.1818%
Kappa statistic	0.6019	
Mean absolute error	0.3271	
Root mean squared error	0.39	
Relative absolute error	69.4995%	
Root relative squared error	80.4455%	
Total Number of Instances	77	

Rotation Forest:

Correctly Classified Instances	63	81.8182%
Incorrectly Classified Instances	14	18.1818%
Kappa statistic	0.6074	
Mean absolute error	0.2462	
Root mean squared error	0.3545	
Relative absolute error	52.3201%	
Root relative squared error	73.1244%	
Total Number of Instances	77	

Stacking:

Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0	
Mean absolute error	0.4707	
Root mean squared error	0.4848	
Relative absolute error	100%	
Root relative squared error	100%	
Total Number of Instances	77	

Stacking C:

Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0	
Mean absolute error	0.4705	
Root mean squared error	0.4848	
Relative absolute error	99.976%	
Root relative squared error	99.999%	
Total Number of Instances	77	

Threshold Selector:

Correctly Classified Instances	52	67.5325%
Incorrectly Classified Instances	25	32.4675%
Kappa statistic	0.3568	
Mean absolute error	0.3121	
Root mean squared error	0.5159	
Relative absolute error	66.3138%	
Root relative squared error	106.422%	
Total Number of Instances	77	

Vote:

Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0	
Mean absolute error	0.4707	
Root mean squared error	0.4848	
Relative absolute error	100%	
Root relative squared error	100%	
Total Number of Instances	77	

Misc:

Hyper Pipes:

Correctly Classified Instances	50	64.9351%
Incorrectly Classified Instances	27	35.0649%
Kappa statistic	0.3468	
Mean absolute error	0.4925	
Root mean squared error	0.4927	
Relative absolute error	104.633%	
Root relative squared error	101.617%	
Total Number of Instances	77	

VFI:

Correctly Classified Instances	63	81.8182%
Incorrectly Classified Instances	14	18.1818%
Kappa statistic	0.618	
Mean absolute error	0.3601	
Root mean squared error	0.4537	
Relative absolute error	76.5165%	
Root relative squared error	93.5814%	
Total Number of Instances	77	

Rules:

Conjunctive Rule:

Correctly Classified Instances	59	76.6234%
Incorrectly Classified Instances	18	23.3766%
Kappa statistic	0.4953	
Mean absolute error	0.2991	
Root mean squared error	0.4343	
Relative absolute error	63.5447%	
Root relative squared error	89.5814%	
Total Number of Instances	77	

Decision Table:

Correctly Classified Instances	59	76.6234%
Incorrectly Classified Instances	18	23.3766%
Kappa statistic	0.5022	
Mean absolute error	0.3198	
Root mean squared error	0.4112	
Relative absolute error	67.9443%	
Root relative squared error	84.8189%	
Total Number of Instances	77	

DTNB:

Correctly Classified Instances	60	77.9221%
Incorrectly Classified Instances	17	22.0779%
Kappa statistic	0.52	
Mean absolute error	0.268	
Root mean squared error	0.413	
Relative absolute error	56.9399%	
Root relative squared error	85.1864%	
Total Number of Instances	77	

J Rip:

Correctly Classified Instances	61	79.2208%
Incorrectly Classified Instances	16	20.7792%
Kappa statistic	0.5575	
Mean absolute error	0.2714	
Root mean squared error	0.4147	
Relative absolute error	57.6578%	
Root relative squared error	85.5477%	
Total Number of Instances	77	

Total Number of Instances	77	

NNge:

Correctly Classified Instances	59	76.6234%
Incorrectly Classified Instances	18	23.3766%
Kappa statistic	0.4882	
Mean absolute error	0.2338	
Root mean squared error	0.4835	
Relative absolute error	49.668%	
Root relative squared error	99.7278%	
Total Number of Instances	77	

BF Tree:

Correctly Classified Instances	60	77.9221%
Incorrectly Classified Instances	17	22.0779%
Kappa statistic	0.5266	
Mean absolute error	0.2838	
Root mean squared error	0.4247	
Relative absolute error	60.3043%	
Root relative squared error	87.6044%	
Total Number of Instances	77	

OneR:

Correctly Classified Instances	64	83.1169%
Incorrectly Classified Instances	13	16.8831%
Kappa statistic	0.6429	
Mean absolute error	0.1688	
Root mean squared error	0.4109	
Relative absolute error	35.8714%	
Root relative squared error	84.7523%	
Total Number of Instances	77	

Decision Stump:

Correctly Classified Instances	61	79.2208%
Incorrectly Classified Instances	16	20.7792%
Kappa statistic	0.5513	
Mean absolute error	0.2934	
Root mean squared error	0.4129	
Relative absolute error	62.3401%	
Root relative squared error	85.177%	
Total Number of Instances	77	

PART:

Correctly Classified Instances	59	76.6234%
Incorrectly Classified Instances	18	23.3766%
Kappa statistic	0.4953	
Mean absolute error	0.2639	
Root mean squared error	0.4523	
Relative absolute error	56.0758%	
Root relative squared error	93.3012%	
Total Number of Instances	77	

FT:

Correctly Classified Instances	60	77.9221%
Incorrectly Classified Instances	17	22.0779%
Kappa statistic	0.5266	
Mean absolute error	0.235	
Root mean squared error	0.4496	
Relative absolute error	49.9202%	
Root relative squared error	92.7267%	
Total Number of Instances	77	

Ridor:

Correctly Classified Instances	58	75.3247%
Incorrectly Classified Instances	19	24.6753%
Kappa statistic	0.4709	
Mean absolute error	0.2468	
Root mean squared error	0.4967	
Relative absolute error	52.4274%	
Root relative squared error	102.460%	
Total Number of Instances	77	

J48:

Correctly Classified Instances	60	77.9221%
Incorrectly Classified Instances	17	22.0779%
Kappa statistic	0.5132	
Mean absolute error	0.2614	
Root mean squared error	0.4241	
Relative absolute error	55.5291%	
Root relative squared error	87.4735%	
Total Number of Instances	77	

Zero R:

Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0	
Mean absolute error	0.4707	
Root mean squared error	0.4848	
Relative absolute error	100%	
Root relative squared error	100%	
Total Number of Instances	77	

J48 Graft:

Correctly Classified Instances	62	80.5195%
Incorrectly Classified Instances	15	19.4805%
Kappa statistic	0.5643	
Mean absolute error	0.2469	
Root mean squared error	0.411	
Relative absolute error	52.4631%	
Root relative squared error	84.7806%	
Total Number of Instances	77	

Trees:

AD Tree:

Correctly Classified Instances	60	77.9221%
Incorrectly Classified Instances	17	22.0779%
Kappa statistic	0.533	
Mean absolute error	0.2804	
Root mean squared error	0.4023	
Relative absolute error	59.585%	
Root relative squared error	82.9842%	

LAD Tree:

Correctly Classified Instances	61	79.2208%
Incorrectly Classified Instances	16	20.7792%
Kappa statistic	0.5513	
Mean absolute error	0.2788	
Root mean squared error	0.4432	
Relative absolute error	59.2295%	
Root relative squared error	91.4243%	
Total Number of Instances	77	

LMT:		
Correctly Classified Instances	62	80.5195%
Incorrectly Classified Instances	15	19.4805%
Kappa statistic	0.5823	
Mean absolute error	0.2575	
Root mean squared error	0.3887	
Relative absolute error	54.7017%	
Root relative squared error	80.1696%	
Total Number of Instances	77	

NB Tree:		
Correctly Classified Instances	53	68.8312%
Incorrectly Classified Instances	24	31.1688%
Kappa statistic	0.3451	
Mean absolute error	0.312	
Root mean squared error	0.5062	
Relative absolute error	66.2947%	
Root relative squared error	104.4028%	
Total Number of Instances	77	

Random Forest:		
Correctly Classified Instances	63	81.8182%
Incorrectly Classified Instances	14	18.1818%
Kappa statistic	0.6074	
Mean absolute error	0.2528	
Root mean squared error	0.3621	
Relative absolute error	53.7151%	
Root relative squared error	74.6926%	
Total Number of Instances	77	

Random Tree:		
Correctly Classified Instances	62	80.5195%
Incorrectly Classified Instances	15	19.4805%
Kappa statistic	0.5765	
Mean absolute error	0.2011	
Root mean squared error	0.4432	
Relative absolute error	42.7343%	
Root relative squared error	91.4109%	
Total Number of Instances	77	

Rep Tree:		
Correctly Classified Instances	58	75.3247%
Incorrectly Classified Instances	19	24.6753%
Kappa statistic	0.4781	
Mean absolute error	0.318	
Root mean squared error	0.4516	
Relative absolute error	67.5629%	
Root relative squared error	93.144%	
Total Number of Instances	77	

Simple Cart:		
Correctly Classified Instances	60	77.9221%
Incorrectly Classified Instances	17	22.0779%
Kappa statistic	0.52	
Mean absolute error	0.2993	
Root mean squared error	0.4339	
Relative absolute error	63.5823%	
Root relative squared error	89.5084%	
Total Number of Instances	77	

User Classifier:		
Correctly Classified Instances	48	62.3377%
Incorrectly Classified Instances	29	37.6623%
Kappa statistic	0	
Mean absolute error	0.4698	
Root mean squared error	0.4848	
Relative absolute error	99.82%	
Root relative squared error	99.999%	
Total Number of Instances	77	

Bibliography

[1] Irish, J. (2004). A 5,500 year old artificial human tooth from Egypt: a historical note. *The International Journal of Oral and Maxillofacial Surgery*, *19*(5), p: 645-647.

[2] Ring, M. (1995). A Thousand Years of Dental Implants: A Definitive History-Part 1. *Compendium of Continuing Education in Dentistry*, *16*(10), p: 1060-1690.

[3] Ring, M. (1995). A Thousand Years of Dental Implants: A Definitive History-Part 2. *Compendium of Continuing Education in Dentistry*, *16*(11), p: 1132-1152.

[4] Brånemark, P. (1959). Vital microscopy of bone marrow in rabbit. *Scandinavian Journal of Clinical and Laboratory Investigation*, *11*(38), p:1-82.

[5] Brånemark, P. (1983). Osseointegration and its experimental background. *The Journal of Prosthetic Dentistry*, *50*(3), p: 399-410. doi:10.1016/S0022-3913(83)80101-2

[6] Mavrogenis, A., Dimitriou, R., Parvizi, J. & Babis, G. (2009). Biology of Implant Osseointegration. *Journal of Musculoskeletal and Neuronal Interactions 9*(2), p: 61-71.

[7] Beckmann, E. (2006). CT scanning the early days. *The British Journal of Radiology*, *79*, p: 5-8. doi:10.1259/bjr/29444122

[8] Hounsfield, G. (1973). Computerized Transverse Axial Scanning (tomography): Part 1. Description of system. *The British Journal of Radiology*, *46*, p: 1016-1022. doi:10.1259/0007-1285-46-552-1016

[9] Schindelin, J., et al. (2012). Fiji: An Open-Source Platform for Biological-Image Analysis. *Nature Methods*, *9*(7), p: 676-682. doi: 10.1038/nmeth.2019

[10] Schneider, C., Rasband, W. & Eliceiri, K. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, *9*(7), p: 671-675. doi:10.1038/nmeth.2089

[11] Marbán, Ó., Mariscal, G. & Segovia, J. (2009). *A Data Mining & Knowledge Discovery Process Model* (p: 438-453). Vienna, Austria: In-Teh.

[12] Holmes, G., Donkin, A. & Witten, I. (1994). WEKA: A Machine Learning Workbench. Proceedings of *Second Australia and New Zealand Conference on Intelligent Information Systems*.

[13] Garner, S., Cunningham, S., Holmes, G., Nevill-Manning, C. & Witten, I. (1995). Applying a Machine Learning Workbench: Experience with Agricultural Databases. Proceedings of *Machine Learning in Practice Workshop, Machine Learning Conference*, p: 14-21.

[14] Zhang, H. (2004). The Optimality of Naive Bayes. Florida Artificial Intelligence Research Society, proceedings of *FLAIRS Conference 2004*, p: 562-567.

[15] Othman, M. & Yau T. (2007). Comparison of Different Classification Techniques Using WEKA for Breast Cancer. International Conference on Biomedical Engineering, proceedings of *IFMBE Conference 2007*, *15*, p: 520-523.